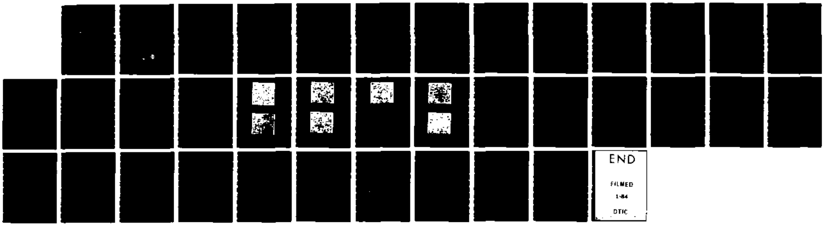
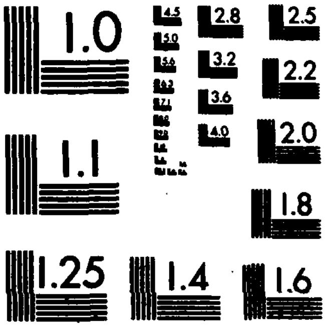
AD-A136 168   MARKOV TEXTURE GENERATION(U) LOUISIANA STATE UNIV BATON   1/1
ROUGE REMOTE SENSING AND IMAGE PROCESSING LAB
R E VASQUEZ-ESPINOSA NOV 82 RSIP/TR-404.82
UNCLASSIFIED   AFOSR-TR-83-1153 AFOSR-81-0112   F/G 9/2   NL




1.0
4.5
5.0
5.6
6.3
7.1
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.1
1.8
1.25
1.4
1.6

MARKOV TEXTURE GENERATION

by

R. E. Vasquez-Espinosa

RSIP TR 404.82

Remote Sensing and Image Processing Laboratory
Department of Electrical and Computer Engineering
Louisiana State University
Baton Rouge, LA 70803

November 1982

This research was supported in part by AFOSR Grant #AFOSR-81-0112

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (*When Data Entered*)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
AFOSR-TR- 83-1153

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (*and Subtitle*)
MARKOV TEXTURE GENERATION

5. TYPE OF REPORT & PERIOD COVERED
TECHNICAL

6. PERFORMING ORG. REPORT NUMBER
RSIP TR #404.82

7. AUTHOR(s)
R.E. Vasquez-Espinosa

8. CONTRACT OR GRANT NUMBER(s)
AFOSR-81-0112

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Remote Sensing & Image Processing Laboratory
Department of Electrical & Computer Engineering
Louisiana State University, Baton Rouge LA 70803

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
PE61102F; 2304/A2

11. CONTROLLING OFFICE NAME AND ADDRESS
Mathematical & Information Sciences Directorate
Air Force Office of Scientific Research /NM
Bolling AFB DC 20332

12. REPORT DATE
NOV 82

13. NUMBER OF PAGES
35

14. MONITORING AGENCY NAME & ADDRESS(*if different from Controlling Office*)

15. SECURITY CLASS. (*of this report*)
UNCLASSIFIED

15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (*of this Report*)
Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (*of the abstract entered in Block 20, if different from Report*)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (*Continue on reverse side if necessary and identify by block number*)

20. ABSTRACT (*Continue on reverse side if necessary and identify by block number*)

To implement the author's theoretical method for defining texture measurements, it is necessary to have the ability to generate textures. In this report, a texture generation method is described based on the use of Markov chains. This method was implemented using a Markov Texture Generation Program.

83 12 19 004

ABSTRACT

To implement our theoretical method for defining texture measurements, it is necessary to have the ability to generate textures. In this report, a texture generation method is described based on the use of Markov chains. This method was implemented using a Markov Texture Generation Program.

1. INTRODUCTION

A theoretical method for defining texture measurements was defined in ~~references 1 and 2.~~ previous works. To implement this method, it is necessary to have the ability to generate textures. Two texture generation procedures are proposed: first, a modified Gagalowicz generation procedure and second, a Markov texture generation procedure [1]. The objective of this report is to describe the implementation of the latter of these generation procedures. Before describing the implementation of the Markov texture generator , it is necessary to point out the importance of the texture generator procedure.

Texture analysis is a very important part of image processing. Textural feature extraction (measurements) is a key point in the conception of image classification schemes. Comparative studies of the efficiency and accuracy of those techniques have been done to identify their advantages [3,4] using real image data. This is reasonable because the ultimate purpose is to analyze real data. Conclusions of such studies might sometimes be not generalizable because of the limited member and variety of the data samples. In order to proceed with comparative studies in a more through manner, one can supplement the real data by generating texture patterns having specified statistics. The characteristics of each technique can be investigated to provide stronger conclusions. In [5], Markov generated textures are considered for a theoretical comparison of four texture algorithms. They examined the amount of texture-context information contained in four algorithms. Texture generation also has important applications such as the generation of test samples for the analysis of the human visual recognition system or for the control of texture classification algorithm [6]. Julesz [7] demonstrated that computer generated images with controlled statistical, topological or heuristics properties are liable to reveal some basic organization principles of information processing in the sensory nervous system. These patterns deprive subjects of their life-long learned habits of recognition and make them rely on more primitive mechanisms. Julesz used Markov textures in his investigation of human texture perception.

The generation of textures, in general, can be divided into two schemes: structural analysis [8-12] and statistical analysis [13-28]. Markov generation procedure belongs to the statistical analysis scheme.

A Markov texture image can be generated using the outcomes of gray tone values from a regular Markov chain by arranging these outcomes in a sequential manner along an image line. In a more formal way, the generation procedure is one which the random field, $X(n,m)$, $0 \leqslant n < \infty$, $0 \leqslant m < \infty$ is generated a row at a time; each row is generated in a left-to-right direction; and the grey level of the left most point of each row is asigned according to an initial distribution. The rest of the points of each row are assigned gray levels according to the one-step transition probabilities, $p(i/j)$, of a Markov chain. The major advantages of this method are a simple generation process and a direct specification of spatial co-occurrence matrix of generated texture.

Markov generation procedure is used in conjuction with design objective 2 [see reference 1]. This is needed because the desires to create a set of measurements which can match the capabilities of spontaneous human texture perception that "resembles" the primitive mechanisms of human perception, and the modification requires to make the Gagalowicz procedure severely restricted the class of texture that can be generated using this method.

In section 2 of this paper, a theory and the nonmeclature is presented. Section 3 describes the implementation procedure. Finally, section 4 presents some sample runs.

2. THEORY

A. General Theory

Definition: Markov chain [29]. A Markov chain is a sequence of experiments performed on a system S with the following properties:

1. At any given time the system is in one of the states E1,E2,.......En and the outcome of each experiment is that S either is unchanged or is changed to a new state from the set {E1,E2,.....En}.
2. The system can change from one state Ei to another state Ej only as a result of the experiment.
3. The probability of changing the system from the state Ei to the state Ej depends only on Ej and on Ei, and the state of the system at the time of the experiment. This probability is denoted by p(j/i).

In a more formal way, consider a stochastic process X={Xn; neN} with a countable state space E. A Markov chain is a sequence of random variables such that

$$p\left\{X_{n+1}=j \,\middle|\, X_0=k,\ldots,X_n=i\right\} = p\left\{X_{n+1}=j \,\middle|\, X_n=i\right\} = p(j/i)$$

$$\text{for } i,j,k \in E$$

That is, the next state Xn+1 is independent of the past states XO,.....Xn-1 provided that the present state Xn can be known. Since p(j/i) gives the probability for the transition of S from state Ei to the state Ej, it is called the transition probability for the Markov chain. It is customary to arrange the p(j/i) into a square array and to call the resulting matrix [p(j/i)] the transition matrix of the Markov chain. The transition matrix has the following properties:

1. The transition matrix represents probabilities. The elements of this matrix p(i,j) are all nonnegative.

$$p(j|i) \geqslant 0 \text{ for any } i, j \in E.$$

2. For each row of the matrix the sum of the elements in that row must be 1.

$$\sum_{i \in E} p(j|i) = 1 \text{ for each } j \in E.$$

Any matrix that satisfies these two condition is called a stochastic matrix.

Theorem [30]: For any n, m, ∈ N with m ≥ 1 and iO,....im ∈ E,

$$p\{X_{n+1}=i_1,\cdots,X_{n+m}=i_m/X_n=i_0\}=p(i_1/i_0)p(i_2/i_1)\cdots\cdots p(i_m/i_{m-1})$$

Corollary [30]: Let TTo be a probability distribution on E, and suppose p{Xo=i}=TTo(i) for all i ∈ E. Then for any m ∈ N and io,....im ∈ E,

$$p\{X_0=i_0,X_1=i_1,\cdots X_m=i_m\}=\Pi_0(i_0)p(i_1/i_0)\cdots\cdots p(i_m/i_{m-1})$$

This corollary shows that the joint distribution of Xo,X1,....Xm is completely specified for every m once the initial distribution TTo and the transition matrix [P] are known.

B. Method of Generating Texture

The textures to be generated are ones which can be represented by random fields generated using Markov chains. It generates a row at a time; each row is generated in a left-to-right directions; and the grey level of the left-most point of each row is assigned according to an initial distribution TTo. The rest of the points of each row are assigned gray levels according to the one-step transition probabilities, p(j/i), of a Markov chain. Using this generation procedures the only parameters neccesary to completely specify the random field X(n,m) are the initial distribution TTo(i), $0 \leqslant i \leqslant 1-1$, and the one-step transition probabilites p(j/i), $0 \leqslant i \leqslant 1-1$, $0 \leqslant j \leqslant 1-1$, where 1 is the number of gray levels. To facilitate the computation of the required expected values, only Markov chains which satisfy the following three constraint conditions are used:

1. Markov chains which have stationary one-step transition probabilities

   $$p(j|i)\,,\ 0 \le i \le l-1\,,\ 0 \le j \le l-1$$

2. Markov chains which have an uniform stationary distribution TTs, i.e.,

   $$\Pi_s(i) = 1/l\,,\ 0 \le i \le l-1$$
   $$\Pi_s(o) = 0 \text{ elsewhere.}$$

3. The initial distribution must be the uniform distribution, i.e.,

   $$\Pi_o(i) = 1/l\,,\ 0 \le i \le l-1$$
   $$\Pi_o(i) = 0 \text{ elsewhere.}$$

The third condition guarantees that TTo = TTs. The resulting two-dimensional random fields have the following properties:

1. For any n,m,k and l, n=k, the random variables X(n,m) and X(k,l) are statistically independent.
2. For any n,m, and l, the random variables X(n,m) and X(k,l) are related by the m-l step transition probabilities. These transition probabilities can be computed directly from the one-step transition probabilities of the Markov chain used to create the random field.
3. The random field X(n,m) is stationary with respect to arbitrary translations.

The following equation relates the Markov chain parameters used to generate a texture with the spatial gray level dependence matrices which can be computed from this texture.

$$E\{S(i,j,\delta)\} = \begin{cases} \Pi_0(i)\, p^d(j|i) \text{ for } \delta=(d,0^\circ) \\ \qquad\qquad\qquad\qquad d=1,2,\ldots\ldots \\ \Pi_0(i)\,\Pi_0(j) \text{ elsewhere} \end{cases}$$

where p (j/i) is the d-step transition probability of the Markov chain.

3. IMPLEMENTATION

The Markov texture generation program is implemented using the ELAS system. This section will presents a description of the program and a user's guide.

A. Program

The program has two main subroutines: MRKV and GMRKV. The first subroutine (MRKV) handles the input data to obtain the information neccesary to generate the Markov texture. The second subroutine (GMRKV) create an output contiguous file that contains the texture generated. This output files is in ELAS format. There are four options for texture patterns that can be generated by the Markov generator program (Fig. 1).

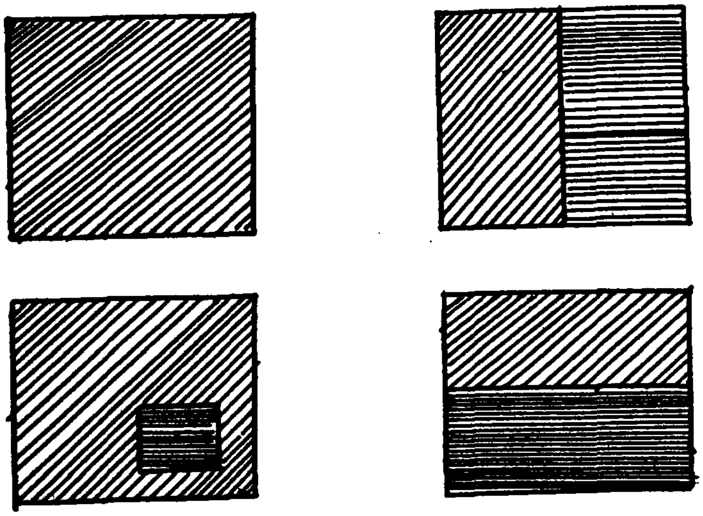


Figure 1. . Four texture options used in Markov generation program.

The subroutine MRKV reads from a given data file the number of elements in state space and the block size of each pattern. The texture options (see fig. 1) are promted to the CRT to be entered by the user. The state space patterns, transition matrices, and initial distribution matrices are read from the given data file in that order. The number of gray levels are equal to the number of elements in state space. Using the number of gray levels and the gray level options, the program computes the shade values. The subroutine GMRKV is called to generates the texture.

Subroutine GMRKV generates a row in a left-to-right direction. The grey level of the left-most point of each row is assigned according to an initial distribution and a uniform random number generator. The rest of the points of each row are assigned grey level according to the transition probabilities (TRANS) and an uniform random number generator

.

Others auxiliary subroutines are needed. Subroutine RANDOM generates a real random number between 0 and 1.0. Subroutine RPALOC and CPALOC are used to allocate an output file in ELAS format.

B. User's Guide

To generate a Markov texture, the user has to be loaded in ELAS. A control file needs to be created. The user has to assign ID1 to the input file that contains the information necessary to generate the texture. The overlay MKRV is yanked and the procedure necessary to generate the texture is started. An example is presented as follow:

```
*ELAS
-what control file?
->MRKV.CF
-file mgr?
->AC MARKOV.DAT ID1
-file mgr?
->LF
1 id1 markov.dat
-file mgr?
->YANK MRKV
-choose one of the following texture options:
- id=0 whole image same texture
- id=1 left half one texture,right half another texture
- id-2 two textures, one in a small box on image
- id=3 top half one texture, bottom half another texture
-
->1
-choose the gray level options:
- ig=0 assign 0 as black
- ig=1 assign 0 as white
```

```
->0
-image file needs to be allocated
-
-input name of file to be allocated
->MARKOV.IMG
-read the block size (n x n) of each pixel
-
->2
-markov generation program termination
-
-file mgr?
->LF
1 idl markov.dat
2 mktx markov.img
-file mgr?
```

To display the texture generated, follow the instructions to display an image given in ELAS manual. In the example above all the capital letter instructions are given by the user. The lower case letter instructions are promted by the program on the CRT.

The input file containing the information necessary is create using the Editor. The format is as follow:

1. Number of elements state space and block size of each patterns. Interger
2. State space patterns. Interger

3. Transition matrices. Real

4. Initial distribution. Real

An example of how the data file has to be created to generate a Markov texture is given as follow:

```
2 2          number of elements state space and block size
1 1 
1 1 
2 2    }     state space patterns
2 2 
0.0 1.0 
1.0 0.0 
0.5 0.5  }    transition matrices
0.5 0.5 
0.33 0.33 
           } initial distribution
0.33 0.33 
```

Appendix A presents a listing of the program. Next section presents several texture generates using this program.

RESULTS

Table 1 shows the transition matrices and the initial distribution matrices for texture A and B. Figures 2 show a texture generated using the transition matrix for texture A, with three space state patterns of a block size of two, three gray level (0 is black), and a pixel size of four.

$$(A)\quad \begin{vmatrix} .8 & 0 & .2 \\ 0 & .4 & .6 \\ .2 & .6 & .2 \end{vmatrix}$$

$$(B)\quad \begin{vmatrix} .2 & .6 & .2 \\ .6 & .4 & 0 \\ .2 & 0 & .8 \end{vmatrix}$$

$$(C)\quad [1/3 \quad 1/3 \quad 1/3]$$

Table 1. The Markov chain parameters used to generate the textures. (A) The one-step transition matrix used to create the texture on the left. (B) The one-step transition matrix used to create the texture on the right. (C) The stationary distribution of both Markov chains.

Figure 3, 4, 5, 6 show texture pairs generated using the transition matrices for texture A and B with three space state patterns of a block size of two, three gray level (0 is black), and a pixel size of one, two, three, and four respectively. The texture option was one.

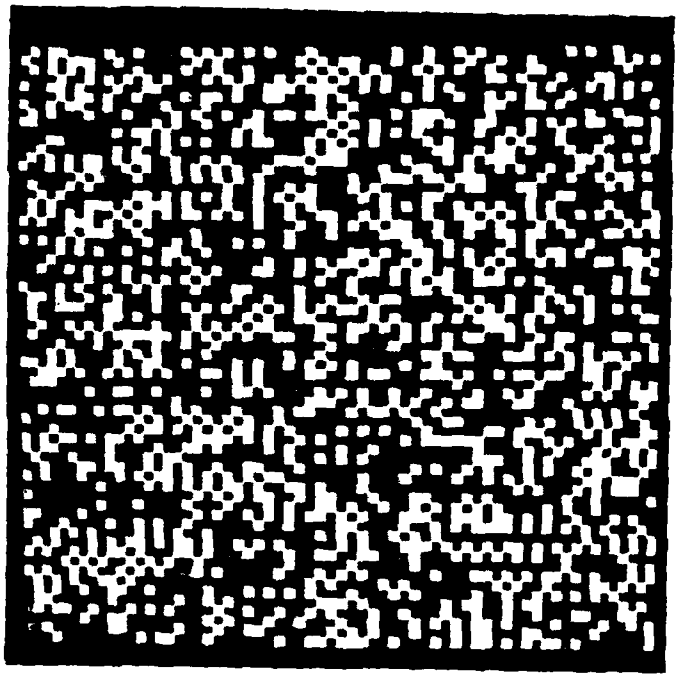


Figure 2. Texture generated using the one-step transition matrix (A), and the stationary distribution matrix (C).

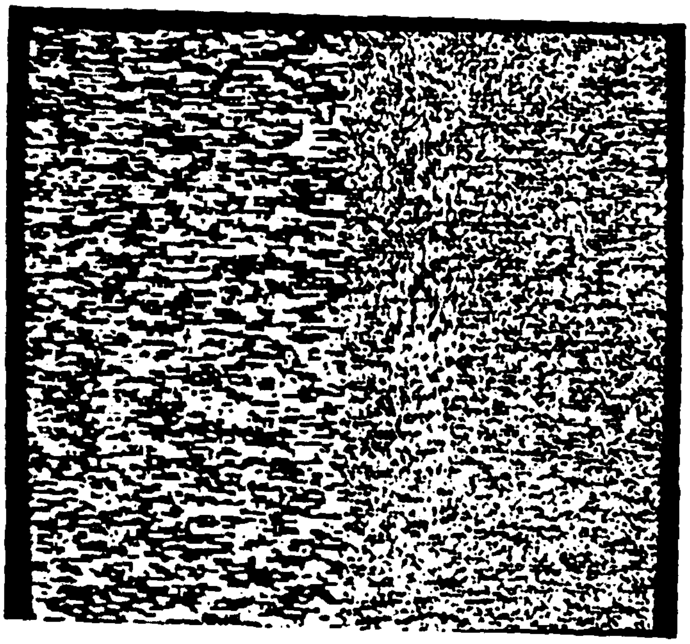


Figure 3. Texture pair generated using the one-step transition matrices (A) and (B), and the stationary distribution matrix (C). Pixel size is one.

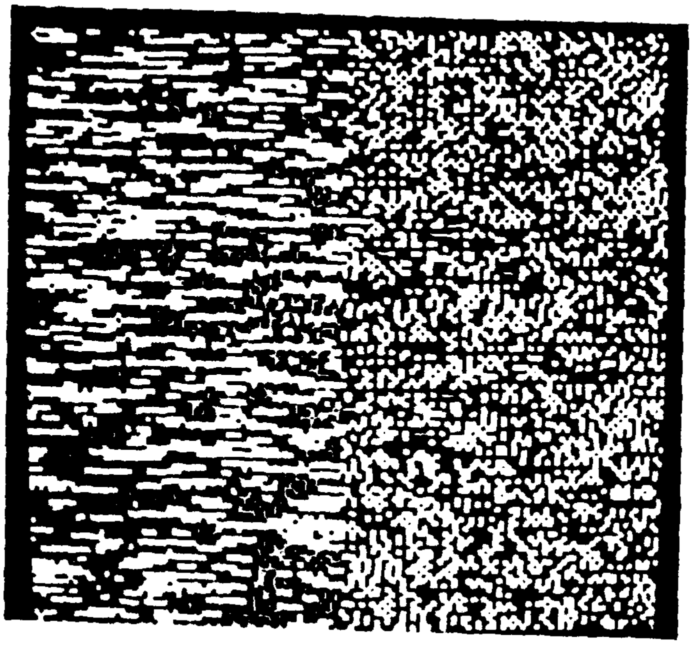


Figure 4. Texture pair generated using the one-step transition matrices (A) and (B), and the stationary distribution matrix (C). Pixel size is two.

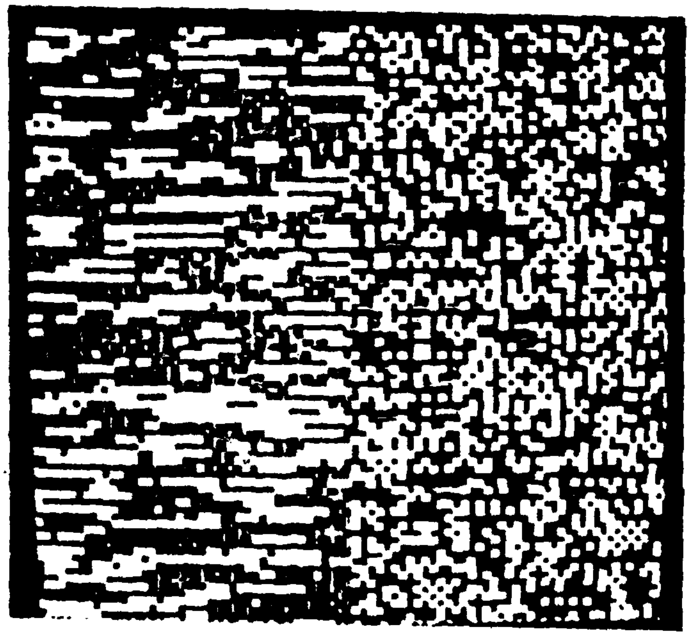


Figure 5. Texture pair generated using the one-step transition matrices (A) and (B), and the stationary distribution matrix (C). Pixel size is three.

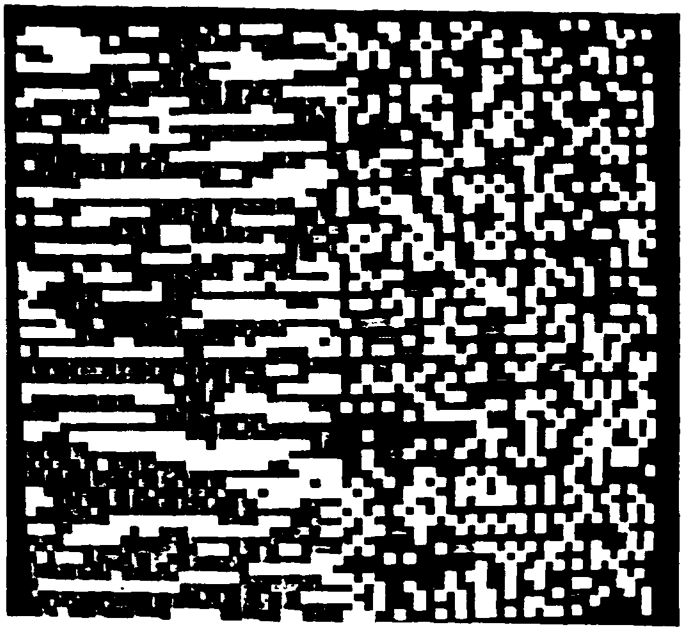


Figure 6. Texture pair generated using the one-step transition matrices (A) and (B), and the stationary distribution matrix (C). Pixel size is four.

Figure 7 and 8 show texture pairs with the same characteristics of the other texture pairs except that the pixel size is two and the texture options are two and three respectively. All this figures demonstrate the ability of our Markov Teture Generation Program.

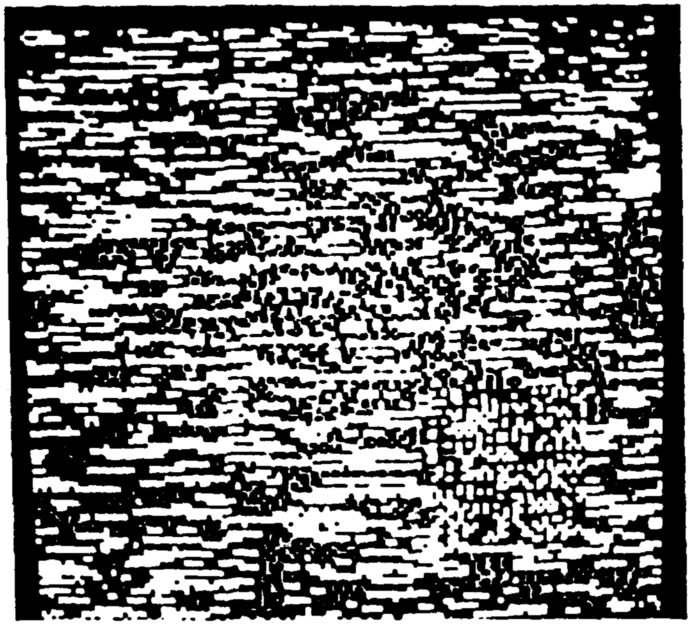


Figure 7. Texture pair generated using the one-step transition matrices (A) and (B), and the stationary distribution matrix (C). Pixel size is two. Texture option is 2.

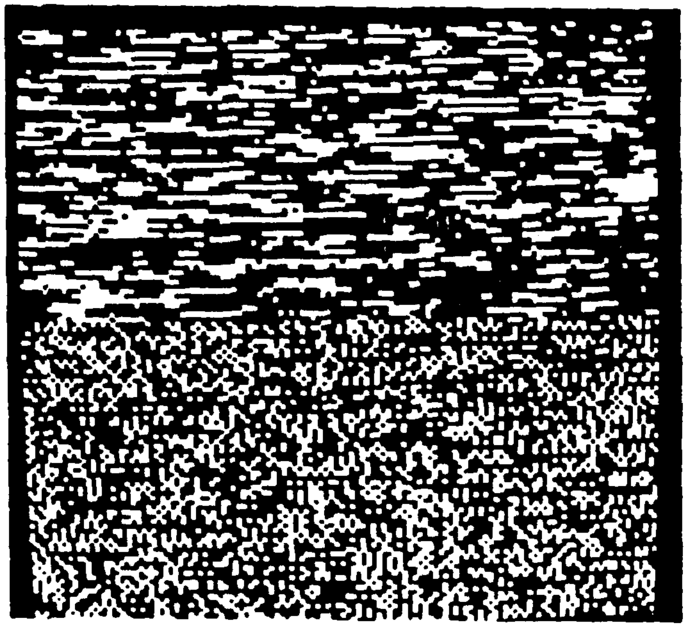


Figure 8. Texture pair generated using the one-step transition matrices (A) and (B), and the stationary distribution matrix (C). Pixel size is two. Texture option is 3.

REFERENCES

1. Conners, R. W. and R. E. Vasquez-Espinosa, "A Theory of Texture Measurement Definition," RSIP Technical Report No. 401.82, Electrical and Computer Engineering Department, Louisiana State University, Baton Rouge, La. 70803

2. Conners, R. W. and R. E. Vasquez-Espinosa, "A Theory of Texture Measurement Definition," IEEE Proceeding 6th International Conference on Pattern Recognition, Munich (Germany), pp. 286-288, 1982.

3. Kirvida, L., and G. Johnson, "Automatic Interpretation of Earth Resources Technology Satellite Data for Forest Management," Symposium on Significant Results Obtained from the Earth Resources Technology Satellite, NASA SP-327, pp. 1976-1982, March 1973.

4. Weszka, J., C. Dyer, and A. Rosenfeld, "A Comparative Study of Texture Measures for Terrain Classification," IEEE Transactions on Systems, Man, and Cybernetics Vol. SMC-6, No. 4, pp. 269-285, April 1976.

5. Conners, R. W. and C. A. Harlow, "A Theoretical Comparison of Texture Algorithms," IEEE Transactions on Pattern

Analysis and Machine Intelligence, Vol. PAMI-2, No. 3, pp. 204-222, May 1980.

6. Monne, J., F. Schmitt, and D. Massaloux, "Bidimensional Texture Synthesis by Markov Chains," Computer Graphics and Image Processing, Vol. 17, pp.1-23, 1981.

7. Julesz, B., "Visual Pattern Discrimination," IRE Transactions on Information Theory, Vol. IT-8, pp.84-92, Feb. 1962.

8. Jayaramamurthy, S. N., "Multilevel Array Grammars for Generating Texture Scenes," IEEE Proceedings Computer Society Conference on Pattern Recognition and Image Processing, Chicago, (Illinois), pp. 391-398, August 1979.

9. Lu, S. Y., and K. S. Fu, "Stochastic Tree Grammar Inference for Texture Synthesis and Discrimination," Computer Graphic and Image Processing, Vol. 9, pp.246-266, 1979.

10. Yokoyama, R. and R. Haralick, "Texture Synthesis using a Growth Model," Computer Graphics and Image Processing, Vol. 8, pp. 369-381, 1978.

11. Ahuja, N., and A. Rosenfeld, "Mosaic Models for Texture," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-3, No.1, pp. 1-11, Jan. 1981.

12. Zucker, S. W., "Towards a Model of Texture," Computer Graphics and Image Processing, Vol. 5, pp. 190-202, 1976.

13. McCormick, B. H., and S. N. Jayaramamurthy, "Time Series Model for Texture Synthesis," Comput. Inform. Sci., Vol. 3, pp. 329-343, 1974.

14. Caelli, T., and B. Julesz, "On Perceptual Analyzers Underlying Visual Texture Discrimination: Part I," Biol. Cybernet., Vol. 28, pp. 167-175, 1978.

15. Caelli, T., and B. Julesz, "On Perceptual Analyzers Underliying Visual Texture Discrimination: Part II," Biol. Cybernet., Vol. 29, pp. 201-214, 1978.

16. Gagalowicz, A., "A New Method for Texture Field Synthesis: Some Applications to the Study of Human Vision," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-3, No. 5, pp. 520-533, Sept. 1981.

17. Julesz, B., E. N. Gilbert, and J. D. Victor, "Visual Discrimination of Textures with Identical Third-Order Statistics," Biol. Cybernet., Vol. 31, pp. 137-140, 1978.

18. Julesz, B., and T. Caelli, "On The Limits of Fourier Decomposition in Visual Textures Perception," Perception,

Vol. 8, pp.69-73, 1979.

19. Julesz, B., E. N. Gilbert, L. A. Shepp, and H. L.Frisch, "Inability of Humans to Discriminate Between Visual Textures that Agree in Second Order Statistics-Revisited," Perception, Vol. 2, pp. 391-405, 1973.

20. Rosenfeld, A., and B. S. Lipkin, "Texture Synthesis," in Picture Processing and Psychopictoris, Academic Press, New York, pp. 309-345, 1970.

21. Mayhew, J. E. W., and J. P. Frisley, "Suprathreshold Contrast Perception and Complex Random Textures," Vision Res., Vol. 18, pp. 895-897, 1978.

22. Mitchell, O. R., "Effect of Spatial Frequency on the Visibility of Unstructured Patterns," J. Opt. Soc. Amer., Vol. 66, pp. 327-332, 1976.

23. Pratt, W., O. Faugeras, and A. Gagalowicz, "Visual Discrimination of Stochastics Texture Fields," IEEE Transactions on Systems, Man, and Cybernetics, Vol.SMC-8, No. 11, pp. 796-804, 1978.

24. Pollack, I., "Discrimination of Third-Order Markov Constrains within Visual Displays," Perception and Psychophys.,

Vol. 13, pp. 276-280, 1973.

25. Purks, S. R., and W. Richards, "Visual Texture Discrimination Using Random-Dot Patterns," J. Opt. Soc. Amer., Vol. 67, pp. 765-771, 1977.

26. Rosenblatt, M., and D. Slepian, "Nth Order Markov Chains with Every N Variable Independent," J. Soc. Indust. Appl. Math., Vol. 10, pp. 537-549, 1962.

27. Yokoyama, R. and R. M. Haralick, "Texture Pattern Images Generation by Regular Markov Chain," Pattern Recognition, Vol. 11, pp. 225-233, 1979.

28. Hassner, M., and J. Sklansky, "The Use of Random Fields as Models of Texture," Computer Graphics and Image Processing, Vol. 12, pp. 357-370, 1980.

29. Goodman, A. W., and J. S. Ratti, Finite Mathematics with Application, Chapter 8, The Macmillan Company, N. Y., 1965.

30. Cinlar, E., Introduction to Stochastic Processes, Chapter 5, Prentice-Hall, 1975.

APPENDIX A

PROGRAMS LISTING

TMDS SUBROUTINE MRKV

Language: Fortran VII, using TMDS subroutines; for Perkin-Elmer 8/32.

Purpose : To handle the input data to obtain the information neccesary to generate Markov Textures.

In/Output : Respectively, the number of elements state space, block size of each patterns, state space patterns, transition matrices, and initial distribution.

Usage : CALL MRKV

FILENAME = LSU1:MRKV.FTN

```
$BATCH
C
C
C    TEXTURE MEASUREMENT DEFINITION SYSTEM
C
C    REMOTE SENSING AND IMAGE PROCESSING LABORATORY
C
C    ELECTRICAL AND COMPUTER ENGINEERING DEPARTMENT
C
C    LOUISIANA STATE UNIVERSITY. BATON ROUGE. 70803
C
C----------------------------------------------------------------
C
C    PROGRAM MRKV
C
C
C    AUTHOR
C
C         R. E. VASQUEZ-ESPINOSA
C
C    DATE
C
C         OCTOBER 1982
C
C---------------------------------------------------------------
      SUBROUTINE BEGIND
      CALL MRKV
      RETURN
      END
C
C
C   SUBROUTINE MRKV
C
C
C
      SUBROUTINE MRKV
      INTEGER STATES(32,32,8),CB(10)
      DIMENSION TRANS(8,8,2),PROB(8,2),ISHADE(64)
      DATA INPUT/'ID1'/,NCFSF/'FMGR'/
C
C   GET LOGIC UNIT FOR THE INPUT FILE
C
      CB(1)=INPUT
      CALL CFSUB(8,CB,CB)
      IF(CB(2) .LE. 0) GO TO 2000
      LUS=CB(2)
C
C   GET LOGIC UNIT FOR THE PRINT FILE
C
      CALL PRF(LUPRF)
C
C   READ NUMBER OF ELEMENTS IN STATE SPACE (NSTATE) AND
C   READ BLOCK SIZE OF EACH PATTERN (NSIZE)
C
      READ(LUS,*)NSTATE,NSIZE
      WRITE(LUPRF,100) NSTATE,NSIZE
  100 FORMAT(10X,'NSTATE= ',I4,2X,'NSIZE= ',I4,/)
```

```
C
C  CHOOSE TEXTURE OPTIONS
C
      WRITE(6,101)
      READ (5,102) ID
C
C  DETERMINE NUMBER OF TEXTURES INVOLVED
C
      IF(ID .EQ. 0) NTEXT=1
      IF(ID .GT. 0) NTEXT=2
C
C  READ STATE SPACE PATTERNS (STATES) FROM THE INPUT DATA.
C
      DO 10 L=1,NSTATE
         DO 10 I=1,NSIZE
            READ(LUS,*)(STATES(I,J,L)  ,J=1,NSIZE)
            WRITE(LUPRF,103)(I,J,L,STATES(I,J,L),J=1,NSIZE)
  103       FORMAT(10X,'STATES (',I2,',',I2,',',I2,')= ',I4)
   10 CONTINUE
C
C  CHECK FOR ZERO CONTENT.
C
      DO 20 L=1,NSTATE
         DO 20 I=1,NSIZE
            DO 15 J=1,NSIZE
               IF(STATES(I,J,L)) 15,30,15
   15       CONTINUE
   20    CONTINUE
      GO TO 40
   30 DO 50 L=1,NSTATE
         DO 50 I=1,NSIZE
            DO 50 J=1,NSIZE
               STATES(I,J,L)=STATES(I,J,L) + 1
   50 CONTINUE
   40 CONTINUE
C
C  READ TRANSITION MATRICES (TRANS) FROM THE INPUT DATA FILE.
C
      DO 60 L=1,NTEXT
         DO 60 I=1,NSTATE
            READ(LUS,*)(TRANS(I,J,L),J=1,NSTATE)
            WRITE(LUPRF,104)(I,J,L,TRANS(I,J,L),J=1,NSTATE)
  104       FORMAT(10X,'TRANS(',I2,',',I2,',',I2,')= ',F10.5)
   60 CONTINUE
      DO 70 L=1,NTEXT
         DO 70 I=1,NSTATE
            DO 80 J=2,NSTATE
               TRANS(I,J,L)=TRANS(I,J,L) + TRANS(I,J-1,L)
   80       CONTINUE
         TRANS(I,NSTATE,L)=1.0
         WRITE(LUPRF,104)(I,J,L,TRANS(I,J,L),J=1,NSTATE)
   70 CONTINUE
C
C  READ THE INITIAL DISTRIBUTION (PROB) FROM THE INPUT DATA FILE.
C
      DO 90 L=1,NTEXT
         READ(LUS,*)(PROB(J,L),J=1,NSTATE)
```

```
            WRITE(LUPRF,105)(J,L,PROB(J,L),J=1,NSTATE)
  105       FORMAT(10X,'PROB(',I2,',',I2,')= ',F10.5)
   90 CONTINUE
      DO 95 L=1,NTEXT
         DO 94 J=2,NSTATE
            PROB(J,L)=PROB(J,L) + PROB(J-1,L)
   94    CONTINUE
            WRITE(LUPRF,105)(J,L,PROB(J,L),J=1,NSTATE)
   95 CONTINUE
C
C  CHOOSE THE GRAY LEVEL OPTION AND COMPUTE THE SHADE VALUES.
C
      NGRAY=NSTATE
      WRITE(6,106)
      READ (5,107) IG
C
C  INITIAL VALUE FOR SHADE VALUES.
C
      N=NGRAY -1
      ISHADE(1)=0
      IF(IG .EQ.  0) IC=1
      IF(IG .EQ.  1) IC=-1
      IF(IG .EQ.  1) ISHADE(1)=N
      DO 96 I=1,N
         ISHADE (I + 1)= ISHADE (I) + IC
  112    FORMAT(10X,'ISHADE(',I3,')= ',I4)
   96 CONTINUE
C
C  COMPUTE SHADE VALUES.
C
      FACTOR=255.0/N
      DO 97 I=1,NGRAY
         ISHADE(I)=ISHADE (I) * FACTOR
         WRITE(LUPRF,112)I,ISHADE(I)
   97 CONTINUE
C
C  GENERATE IMAGE.
C
      CALL GMRKV(NSTATE,NSIZE,PROB,TRANS,ID,STATES,ISHADE)
      WRITE(6,108)
C
C  RETURN CONTROL TO FILE MANAGER (FMGR).
C
      CB(1)=NCFSF
      CALL CFSUB(11,CB,CB)
      RETURN
 2000 WRITE(6,110)
C
C FORMAT AND ERROR STATEMENT.
C
  101 FORMAT('&CHOOSE ONE OF THE FOLLOWING TEXTURE OPTIONS:',/
     *'& ID= 0 WHOLE IMAGE SAME TEXTURE',/
     *'& ID= 1 LEFT HALF ONE TEXTURE, RIGHT HALF ANOTHER TEXTURE',/
     *'& ID= 2 TWO TEXTURES, ONE IN A SMALL BOX ON IMAGE',/
     *'& ID= 3 TOP HALF ONE TEXTURE, BOTTOM HALF ANOTHER TEXTURE',/
     *)
  102 FORMAT(I1)
 106 FORMAT('&CHOOSE THE GRAY LEVEL OPTIONS: ',/
    *'& IG= 0 ASSIGN 0 AS BLACK',/
    *'& IG= 1 ASSIGN 0 AS WHITE',/)
 107 FORMAT(I1)
 108 FORMAT('&MARKOV GENERATION PROGRAM TERMINATION'/)
 110 FORMAT('&****ERROR WITH LOGIC UNIT FOR INPUT FILE****',/)
     RETURN
     END
$BEND
```

TMDS SUBROUTINE QMRKV

Language : Fortran VII, using TMDS subroutines; for Perkin-Elemer 8/32.

Purpose : To generate Markov textures.

In/Output : Respectively, the input from the subroutine MRKV. The output is a contiguous file that contains the texture generated in Elas format.

Usage : CALL QMRKV(NSTATE,NSIZE,PROB,TRANS,ID,STATES,
IDSHADE).

NSTATE - input state space elements
NSIZE - input block size of patterns
PROB - input initial distribution.
TRANS - input transition matrices.
ID - input texture options.
STATES - input state space patterns.
IHSHADE- input gray level values.

Program Logic : It generates a row at a time; each row is generated in a left-to-right directions; and the grey level of the left-most point of each row is assigned according to an initial distribution. The rest of the points of each row are assigned gray level values according to the one-step transition probabilities (TRANS).

FILENAME = LSU1:QMRKV.FTN

```
C
C
C   TEXTURE MEASUREMENT DEFINITION SYSTEM
C
C   REMOTE SENSING AND IMAGE PROCESSING LABORATORY
C
C   ELECTRICAL AND COMPUTER ENGINEERING DEPARTMENT
C
C   LOUISIANA STATE UNIVERSITY. BATON ROUGE. 70803
C
C-----------------------------------------------------------------------
C
C   PROGRAM QMRKV
C
C
C   AUTHOR
C
C        R. E. VASQUEZ-ESPINOSA
C
C   DATE
C
C        OCTOBER 1982
C
C---------------------------------------------------------------------
C
C
C
      SUBROUTINE QMRKV(NSTATE,NSIZE,PROB,TRANS,ID,STATES,
     *ISHADE)
      INTEGER STATES(32,32,8)
      DIMENSION IB(512),TRANS(8,8,2),PROB(8,2),KBUFF(512),
     *ISHADE(64)
      LOGICAL*1 PIXEL(512),LBYT(4)
      EQUIVALENCE (IP,LBYT)
      DATA NAME/'MKTX'/
C
C
      CALL PRF(LUPRF)
C
C
      WRITE(LUPRF,9300)NSTATE,NSIZE
 9300 FORMAT(10X,'NSTATE= ',I3,2X,'NSIZE= ',I3)
      DO 9100 L=1,NSTATE
      DO 9100 I=1,NSIZE
      DO 9100 J=1,NSIZE
         WRITE(LUPRF,9200) I,J,L,STATES(I,J,L)
 9200    FORMAT(10X,'STATES(',I2,',',I2,',',I2,') =',I4)
 9100 CONTINUE
C  INITIAL VARIABLES
C
      IL= 1
      LL= 512
      IE= 1
      LE= 512
      NBR= 512
      NC= 1
      IRAN= 123456789
```

```
C
C  CHECK IF THE IMAGE FILE IS ALREADY ALLOCATED.
C
      IB(1)=NAME
      CALL CFSUB(8,IB,IB)
      LU=IB(2)
      IF(LU .GT. 0)GO TO 200
      WRITE(6,100)
      IB(1)=NAME
      IB(3)=IL
      IB(4)=LL
      IB(5)=IE
      IB(6)=LE
      IB(7)=NC
      CALL RPALOC(LU,NAME,IB)
      IF(LU .LT. 0)GO TO 9000
      IB(1)= NAME
      IB(4)= LL
      IB(3)= IL
      IB(5)= IE
      IB(6)= LE
      IB(7)= NC
      CALL ELTRAN (LU,4,0,200,IB,IST)
      GO TO 400
C
C  FILE WAS PREVIOUSLY ALLOCATED, CHECK FOR CORRECT SIZE.
C
  200 CALL ELTRAN(LU,3,0,200,IB,IST)
      IF(IB(1) .NE. NAME) GO TO 9002
      IF(IB(3) .NE. IL) GO TO 9004
      IF(IB(4) .NE. LL) GO TO 9006
      IF(IB(5) .NE. IE) GO TO 9008
      IF(IB(6) .NE. LE) GO TO 9010
      IF(IB(7) .NE. NC) GO TO 9012
  400 CONTINUE
C
C  LET READ THE BLOCK SIZE OF EACH PIXEL
C
      WRITE(6,101)
      READ (5,102) NN
      I1=0
      IN=512/NN
      DO 30 IROW=1,IN
         IF(I1 .EQ. NSIZE) I1=0
         I1=I1 + 1
         IF(I1 .NE. 1) GO TO 40
C
C  DETERMINE CURRENT STATE FOR IMAGE GENERATION.
C
         IX=IRAN
         CALL RANDOM (IX,IRAN,XRAN)
         DO 50 I=1,NSTATE
            IF(XRAN .LE. PROB(I,1)) GO TO 60
   50    CONTINUE
   60    ICURNT= I
         NB= NSIZE * NN
         DO 70 J=1,512,NB
```

```
            JCOL= J
C
C  DETERMINE TEXTURE INDEX.
C
            INDX= 1
            IF (ID .EQ. 0) GO TO 71
            IA = IROW * NN
            IF(ID .EQ. 1 .AND. JCOL .GT. 256) GO TO 72
            IF(ID .EQ. 3 .AND. IA .GT. 256) GO TO 72
            IF(ID .NE. 2) GO TO 71
            IF(ID .LE. 320 .OR. IA .GT. 448) GO TO 71
            IF(JCOL .LE. 320 .OR. JCOL .GT. 448) GO TO 71
   72       INDX=2
   71       CONTINUE
C
C  DETERMINE CURRENT STATE.
C
            IX = IRAN
            CALL RANDOM (IX, IRAN, XRAN)
            DO 80 K= 1, NSTATE
               IF(XRAN .LE. TRANS(ICURNT, K, INDX)) GO TO 90
   80       CONTINUE
   90       ICURNT= K
            KBUFF(J)= K
   70    CONTINUE
C
C  GENERATE EACH LINE (512 PIXELS).
C
   40    DO 205 JLK=1, 512, NB
            JMAX= JLK + NB - 1
            IF(JMAX .GT. 512) JMAX=512
            J2= NN
            K1= KBUFF(JLK)
            DO 210 K= JLK, JMAX
               J1=J2/NN
CCCCCCCCCCCCCCCWRITE(LUPRF, 9200)I1, J1, K1, STATES(I1, J1, K1)CCCCCCCCCCCCCC
               M= STATES (I1, J1, K1)
               IP= ISHADE (M)
CCCCCCCCCCCCCCCWRITE(LUPRF, 111)CIROW, JLK, K, M, IPCCCCCCCCCCCCCCCCCCCCCCCCC
CC111CCCCCCCCCCFORMAT(5X, 'IROW=C', I3, 2X, 'JLK=C', I3, 2X, 'K=C', I3, CCCCCCCCC
CCCCC*CCCCCCCCC2X, 'M=C', I3, 2X, 'IP=C', I4)CCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
               PIXEL(K)= LBYT(4)
               J2= J2 + 1
  210       CONTINUE
  205    CONTINUE
C
C  PUT THE LINE IN THE OUTPUT FILE.
C
         NNX=NN
         DO 220 J = 1, NN
            I = IROW * NN - NNX + 1
            CALL RDWR(LU, 4, I, 1, IB, PIXEL, 1)
            NNX= NNX - 1
  220    CONTINUE
   30 CONTINUE
       RETURN
C
```

```
C  ERROR STATEMENTS.
C
 9000 WRITE(6,9001)
 9001 FORMAT('&***ERROR IN ALLOCATING THE OUTPUT FILE',/)
      RETURN
 9002 WRITE(6,9003)
 9003 FORMAT('&***ERROR USAGE NAME NOT MATCH',/)
      RETURN
 9004 WRITE(6,9005)
 9005 FORMAT('&***ERROR INITIAL LINE NOT MATCH',/)
      RETURN
 9006 WRITE(6,9007)
 9007 FORMAT('&***ERROR LAST LINE NOT MATCH',/)
      RETURN
 9008 WRITE(6,9009)
 9009 FORMAT('&***ERROR INITIAL ELEMENT NOT MATCH',/)
      RETURN
 9010 WRITE(6,9011)
 9011 FORMAT('&***ERROR LAST ELEMENT NOT MATCH',/)
      RETURN
 9012 WRITE(6,9013)
 9013 FORMAT('&***ERROR CHANNEL NUMBER NOT MATCH',/)
      RETURN
C
C  FORMAT STATEMENT.
C
  100 FORMAT('&IMAGE FILE NEED TO BE ALLOCATED',/)
  101 FORMAT('&READ THE BLOCK SIZE (N x N) OF EACH PIXEL',/)
  102 FORMAT(I1)
      END
```